REMARKS/ARGUMENTS

The amendments to independent claims 31, 40, 48, and 51 do not require further searching by the Examiner. The amendments correct a grammatical error. Specifically, the singular subject "each" has a verb "have" intended for a plural subject. Thus, "have" is amended to "has".

The Examiner rejects Claims 31-56 under 35 U.S.C.§103(a) as being unpatentable over Wan, et al., in view of Pruthi, et al., and further in view of Fuh, et al.

The cited prior art fails to teach or suggest at least the following italicized features of the pending independent claims:

> 31. A method for identifying a corresponding session for a packet, comprising:
>
> (a) in a first session, a *first endpoint transmitting first and second sets of packets, respectively, to a session monitor and a second endpoint, wherein the first and second sets of packets have differing information, wherein each packet in the first set of packets is used for determining network performance information, and wherein each of the first and second endpoints has an associated electronic address on a network and a session identifier*;
>
> (b) the session monitor receiving at least a first packet in the first packet set, the first packet comprising at least the network address and session identifier associated with the first endpoint;
>
> (c) determining whether at least one of the first endpoint's network address and session identifier correspond to an active session entry recorded in a first set of data structures, *the first set of data structures comprising active session entries, each entry in the first set of data structures having at least network addresses for each of the endpoints to the corresponding session*;
>
> (d) *when at least one of the first endpoint's network address and session identifier correspond to an active session entry in the first set of data structures*, updating the corresponding entry to include the network performance information associated with the at least a first packet;
>
> (e) *determining whether at least one of the first endpoint's network address and session identifier correspond to an active session entry recorded in a second set of data structures, the second set of data structures having active session entries, each of the entries in the second set of data structures failing to comprise network addresses for each of the endpoints to the corresponding session*; and
>
> (f) *when at least one of the first endpoint's network address and session identifier correspond to an active session entry in the second set of data structures*, updating the entry to include the performance information associated with the at least a first packet.

40. In a network, the network comprising:

(i) a session monitor operable to track network performance for a plurality of sessions; and

(ii) *first endpoint and second endpoints, the first endpoint being operable to transmit first and second sets of packets, respectively, to the session monitor and the second endpoint, wherein the first and second sets of packets have differing information, wherein each packet in the first set of packets is used by the session monitor to determine network performance information, and wherein each of the first and second endpoints has an associated electronic address on a network and a session identifier*, the session monitor comprising:

(a) an input operable to receive at least a first packet in the first packet set, the first packet comprising at least the network address and session identifier associated with the first endpoint; and

(b) a matcher operable to:

(b1) determine whether at least one of the first endpoint's network address and session identifier correspond to an active session entry recorded in a first set of data structures, *the first set of data structures comprising active session entries, each entry in the first set of data structures having at least network addresses for each of the endpoints to the corresponding session*;

(b2) *when at least one of the first endpoint's network address and session identifier correspond to an active session entry in the first set of data structures*, update the corresponding entry to include the performance information associated with the at least a first packet;

(b3) *determine whether at least one of the first endpoint's network address and session identifier correspond to an active session entry recorded in a second set of data structures, the second set of data structures having active session entries, each of the entries in the second set of data structures failing to comprise network addresses for each of the endpoints to the corresponding session*; and

(b4) *when at least one of the first endpoint's network address and session identifier correspond to an active session entry in the second set of data structures*, update the entry to include the performance information associated with the at least a first packet.

48. In a network, the network comprising:

(i) a session monitor operable to track network performance for a plurality of sessions; and

(ii) first endpoint and second endpoints, *the first endpoint being operable to transmit first and second sets of packets, respectively, to the session monitor and the second endpoint, wherein the first and second sets of packets have differing information, wherein each packet in the first set of packets is used by the session monitor to determine network performance information, and wherein each of the first and second endpoints*

has an associated electronic address on a network and a session identifier, a method comprising:

(a) the first endpoint receiving at least a first packet communicated between the first endpoint and a second endpoint to a first session, the first packet comprising an address of the first endpoint on the network, an address of the second endpoint on the network, *and voice information, and being associated with the second packet set*; and

(b) *the first endpoint transmitting at least a second packet to a session monitor, the at least a second packet including the respective first and second network addresses of the first and second endpoints and being associated with the first packet set.*

51. In a network, the network comprising:

(i) a session monitor operable to track network performance for a plurality of sessions; and

(ii) first endpoint and second endpoints, *the first endpoint being operable to transmit first and second sets of packets, respectively, to the session monitor and the second endpoint, wherein the first and second sets of packets have differing information, wherein each packet in the first set of packets is used by the session monitor to determine network performance information, and wherein each of the first and second endpoints has an associated electronic address on a network and a session identifier*, the first endpoint comprising:

(a) an input operable to receive at least a first packet communicated between the first and second endpoints to a first session, the first packet comprising a network address of the first endpoint, a network address of the second endpoint, and voice information, and being associated with the second packet set; and

(b) *a transmitter operable to transmit at least a second packet to a session monitor, the at least a second packet including the respective first and second network addresses of the first and second endpoints and being associated with the first packet set.*

54. A session packet for transmission on a network, comprising:

a source network address of a first participant to a *Voice over Internet Protocol (VoIP) session*;

a destination network address associated with a session monitor;

a network address of a second participant to the VoIP session; and

session information associated with the *VoIP* session.

In distinct contrast, Wan, et al., are directed to an architecture for reducing congestion of real time data traffic on a multimedia communications network having a traffic control mechanism. The method includes the step of first extracting from data traffic in the multimedia communications network information regarding congestion of the multimedia communications

network. This extraction is performed by a network of monitors. Secondly, congestion is regulated by a central server that receives network information from the monitors and uses the information to analyze congestion status and communicate instructions to the multimedia communications network to reduce congestion.

Wan, et al., disclose the use of RTCP packets transmitted selectively to monitors. The monitors forward the packets to a call admission control module that uses the network performance information in the packets to detect congestion in the network. When congestion is detected, relevant gatekeepers are informed.

Wan, et al., are entirely silent regarding tracking active RTCP sessions to pair up the performance information with the session.

Similarly in contradistinction to the features recited in the independent claims, Pruthi, et al., are directed to a method for monitoring data on a first communication line. Data is received from the first communication line and a plurality of packets are extracted from the data. Statistics are then recursively generated, the statistics corresponding to the plurality of packets. As shown in Figure 1, the network monitor 102 is coupled to the network N1106 via a first communication line 104. The monitor receives (monitors) data communications (traffic) on communication line 104 and provides real-time metrics or statistics of the data traffic on the communication line 104. Packets are extracted from the bit stream and converted into records stored in memory. The records are generated by first determining the type (protocol or layer) of each packet (step 414) and then filtering the packets (step 416) based on their determined types. An index is generated (step 418) for each packet and the packet is then converted into an indexed record (step 420) and stored in memory (step 422). The time when the network monitor received each IP packet is used as an index for each IP packet. Exemplary information retained respecting each packet includes the type of the packet, the size of the packet, a packet number, an interface number, an application, and an associated session. Further statistics are then generated (step 426) using the statistics previously generated for the packets and records are then provided to one or more applications such as a display device (step 428), a router for dynamically adjusting network

routing based on the further statistics (step 430), and a billing service for billing clients based on quality or quantity of service as determined based on the generated statistics (step 432).

Alternatively, the record may include a plurality of fields, each corresponding to a portion of the IP packet such as a source address or destination address, and filtering may be performed based on any one or more of the plurality of fields. Statistics measured include packet size distributions, protocol distributions, bandwidth usage per client, bandwidth usage by domain, average response time per server, average round-trip time between server-client pair, and performance metrics (e.g., the ratio of the number of bits in IP packets received to the number of bits in all packets received for each successive minute).

While Pruthi, et al., discuss in ¶'s 46-48 generating an index corresponding to one or more received packets, Pruthi fails to overcome the deficiencies noted above in relation to Wan, et al. Furthermore, Pruthi, et al., fail to teach the use of first and second sets of data structures to contain network performance information respecting unidentified and identified sessions, respectively. Additionally, Pruthi, et al., fail to teach dual unicasting in which separate packets are transmitted to the other endpoint and a performance monitor. Rather, Pruthi, et al., teach extracting packets being exchanged between session endpoints to avoid intruding into the network to evaluate or estimate network performance. Intrusion by introducing additional packets into the network can further degrade performance. (¶0008.)

Fuh, et al., are directed to a method and apparatus that provide network access control, and to the need for a mechanism allowing users to use remote access via the Internet without requiring advance knowledge of the IP address of the firewall router and without restricting a user to a particular host or client.

To accomplish this need, a network device is configured to intercept network traffic initiated from a client and directed toward a network resource, and to locally authenticate the client. Authentication is carried out by comparing information identifying the client to authentication information stored in the network device. An authentication cache in the network device stores the authentication information. If the client identifying information is authenticated

successfully against the stored authentication information, the network device is dynamically re-configured to allow network traffic initiated by the client to reach the network resource. If local authentication fails, new stored authentication is created for the client, and the network device attempts to authenticate the client using a remote authentication server. If remote authentication is successful, the local authentication information is updated so that subsequent requests can authenticate locally. As a result, a client may be authenticated locally at a router or similar device, reducing network traffic to the authentication server.

Each access control list or ACL is a list of information that firewall router 210 may use to determine whether packets arriving at or sent from a particular interface may be communicated within or outside the firewall router. For example, in an embodiment, input ACL 424 may comprise a list of IP addresses and types of allowable client protocols. Assume that firewall router 210 receives an inbound packet from client 306 at external interface 420 that is intended for target server 222. If the IP address of client 306 is not stored in input ACL 424, then firewall router 210 will not forward the packet further within the circuitry or software of the firewall router. Output ACL 426 similarly controls the delivery of packets from firewall router 210 to resources located outside external interface 420. Input ACL 428 and output ACL 430 govern packet flow to or from internal interface 422.

The ACLs are linked to authentication caches 432, 434. Each authentication cache represents a valid user authentication. Each authentication cache may include a table of hashed entries of information such as a source IP address, a destination IP address, a source port value, a destination port value, and state information.

When received, the packets of the request are examined. For example, when the HTTP request arrives at the external interface 420 of the firewall router 210, Authentication Proxy 400 examines packets of the request. In block 706, the process determines whether a source IP address of the request is found in the standard access control list. For example, Authentication Proxy 400 determines whether the source IP address in the header field of the packets

corresponds to any entry in the filtering mechanism 219 configured in the Authentication Proxy 400.

If the test of block 706 is affirmative, the authentication caches are searched for the source IP address. In block 710, the process tests whether the source IP address is found. For example, if Authentication Proxy 400 determines that the source IP address matches at least one IP address stored in the filtering mechanism 219, then the Authentication Proxy 400 attempts to authenticate the user 302. In the preferred embodiment, Authentication Proxy 400 searches authentication caches 432, 434 for the source IP address. The goal of this search is to determine if the source IP address of the HTTP packet corresponds to an entry in any of the authentication caches 432, 434.

If the source IP address of the HTTP packet from client 306 does not match any of the entries in the filtering mechanism 219, then Authentication Proxy 400 denies passage to the HTTP packet and makes no attempt at authentication, as shown by block 707 of FIG. 7A. As a result, advantageously, the packet is turned away at the interface and never reaches internal software and hardware elements of the firewall router.

After the new authentication cache is created, login information is requested from the client, as shown in block 724. For example, Authentication Proxy 400 obtains authentication information from User 302 by sending a login form to client 306. The login form is an electronic document that requests User 302 to enter username and password information, as shown by path 403.

However, Fuh, et al., fail to overcome the deficiencies noted above in Wan, et al., and Pruthi, et al.; thus, all claims are patentably distinguishable from the cited references.

The Examiner counters that the following asserted patentable distinctions are unclaimed: (a) tracking active RTCP sessions to pair up the performance information with the session and (b) dual unicasting in which separate packets are transmitted to the other endpoint and a performance monitor. The Examiner further counters that, because the claims simply specify first and second sets of data structures to contain network performance information respecting

unidentical and identified sessions and do not require the sets of data structures to be identical or different or separate, they can be subsets of a common set of data structures as taught by Pruthi.

We disagree with the Examiner's statements.

Feature (a) is claimed specifically in dependent claims 36, 37, 45, 50, and 53.

Feature (b) is claimed specifically in independent claims 31, 40, 48, and 51.

Finally, while the Examiner is correct that the pending claims do not require that the first and second sets of data structures be stored in separate memory locations, the claims do require the sets of data structures to contain *different* information. Claim 31, for instance, requires the members of the first set of data structures to have a network address for each participating endpoint while the members of the second set of data structures do *not* have a network address for each participating endpoint. Because the first and second sets of data structures contain different information, they *are* different, even though they may be consolidated together.

Accordingly, the pending claims are allowable.

The dependent claims provide further reasons for allowance.

Dependent claims 32 and 41 require at least some of the packets in the second set of packets to include media information associated with the first session, and, in steps (c) and (e), a corresponding entry to be identified using the network address and session identifier of the first endpoint. As noted, Pruthi, et al., appear to teach the use of ATM session identifier alone. (*See* example at ¶¶ 0047 and 0048.)

Dependent claims 34 and 43 require the session monitor to:

determine whether a pair of session entries in the second set of data structures pertain to a common session; and

when the second set of data structures includes a pair of session entries pertaining to a common session, remove the pair of entries from the second set of data structures and adding the pair of session entries to a common session entry in the first set of data structures. Dependent claims 35 and 44 further require, when the at least one of the first endpoint's network address and session identifier are not in the first and second sets of data structures, the at least one of the

first endpoint's network address and session identifier to be added to the second set of data structures. As noted, Pruthi, et al., fail to teach the use of first and second sets of data structures to contain network performance information respecting unidentified and identified sessions, respectively. (*See also* dependent claims 36-37, 45-46, 50, and 53.)

Dependent claims 49 and 52 require the first endpoint to transmit at least a second packet to a session monitor when a value of a flag has a first predetermined value. As noted above, the flag indicates whether or not the sending, or second, endpoint already sent a packet to the session monitor. Pruthi, et al., teaches away from this step as it teaches the extraction of packets being exchanged between session endpoints to avoid intruding into the network to evaluate or estimate network performance. According to Pruthi et al., intrusion by introducing additional packets into the network can further degrade performance and is therefore highly undesirable. (¶0008.),

Based upon the foregoing, Applicants believe that all pending claims are in condition for allowance and such disposition is respectfully requested. In the event that a telephone conversation would further prosecution and/or expedite allowance, the Examiner is invited to contact the undersigned.

Respectfully submitted,

SHERIDAN ROSS P.C.

By: Douglas Swartz

Douglas W. Swartz
Registration No. 37,739
1560 Broadway, Suite 1200
Denver, Colorado 80202-5141
(303) 863-9700

Date: May 29, 2007